A neuroevolutionary Approach to Feature Selection using Multiobjective Evolutionary Algorithms

Renê S. Pinto

Institute of Polymers and Composites, University of Minho,
Campus de Azurém, 4800-058 Guimarães, Portugal
Email: b8057@dep.uminho.pt

M. Fernanda P. Costa

Centre of Mathematics, University of Minho,
Campus Gualtar, 4710-057 Braga, Portugal
Email: mfc@math.uminho.pt

Lino A. Costa

ALGORITMI Center, University of Minho,
Campus Gualtar, 4710-057 Braga, Portugal
Email: lac@dps.uminho.pt

António Gaspar-Cunha

Institute of Polymers and Composites, University of Minho,
Campus de Azurém, 4800-058 Guimarães, Portugal
Email: agc@dep.uminho.pt

Abstract Feature selection plays a central role in predictive analysis where datasets have hundreds or thousands of variables available. It can also reduce the overall training time and the computational costs of the classifiers used. However, feature selection methods can be computationally intensive or dependent of human expertise to analyze data. This study proposes a neuroevolutionary approach which uses

multiobjective evolutionary algorithms to optimize neural network parameters in order to find the best network able to identify the most important variables of analyzed data. Classification is done through a Support Vector Machine (SVM) classifier where specific parameters are also optimized. The method is applied to datasets with different number of features and classes.

1 Introduction

In predictive analysis, feature selection is the process of identifying the most important, preferably a few, variables or parameters which are relevant in predicting the outcome. Other motivations can exist, such as: feature set reduction, to reduce resource utilization on future data collections; general data reduction, to increase algorithm speed; or performance improvement, to increase predictive accuracy [1]. For a n-dimensional dataset there exist 2^n possible feature subsets, becoming impractical to evaluate all possible solutions for a large n, leading to an NP-Hard combinatorial problem [2].

Several studies have been proposed to tackle feature selection problems. Simultaneously, there is research work using multiobjective evolutionary algorithms (MOEA) applied to different data classifiers. However, according to [3] most of the approaches for feature selection concerning optimization techniques are based on a single objective. There are a few studies which use multiobjective optimization for feature selection problems.

In [4], the authors proposed a framework for SVM based on multiobjective optimization to minimize the risk of the classifier. The same approach is presented in [5] with the aim of minimizing the number of features of the model. In [6], the authors used hierarchical MOEA to perform feature selection by generating a set of classifiers and selecting the best set of them. In [7], a MOEA optimization methodology is proposed to deal with feature selection problems using a SVM classifier. The proposed approach is applied and validated in a problem of cardiac Single Proton Emission Computed Tomography (SPECT).

In [8], [9] and [10] authors apply successfully neuroevolutionary approaches in different kinds of problems concerning multiobjective optimization.

The present study suggests a neuroevolutionary approach to deal with feature selection problems. In order to reduce complexity of the optimization, artificial neural networks (ANNs) are used to map the most relevant features of analyzed data. MOEA is applied to optimize and find the best classifier parameters and ANNs which gives the most relevant features. The methodology is applied in datasets with different numbers of features, samples and classes. To compare the results, a binary approach, i.e., without using ANNs, is also applied.

2 Methodology

Regarding feature selection problems, that usually leads with thousands of features, the binary representation can increase drastically the computational costs necessary to evaluation because the search space increases with the number of features, since each feature is represented as one single bit in the chromosome of genetic algorithm. Usually, bit 0 means that the feature should not be considered by the classifier and bit 1 means the opposite, i.e., feature should be considered in the classification process. Therefore, this study proposes an alternative codification scheme, based on ANNs. Each chromosome encodes the weights and biases of an ANN instead of considering all the binary features for classification. The ANN is structured in three layers, where the Input Layer receives the number of a single feature and the output is the probability of the input feature being considered by the classifier. The number of inputs is the number of bits necessary to encode the number of features. For instance, if a dataset is composed by samples with 2000 features, 11 bits are required. On the other hand, the same example using binary representation it will requires a chromosome of at least 2000 genes to encode each feature. Although this study use a fixed topology for the ANNs (with 20 neurons in the hidden layer), different topologies can be used by the MOEA. Fig. 1 illustrates the ANN considering the topology for the given example. The chromosome (without classifier parameters) will need only 272 genes to encode all ANN parameters instead of 2000 genes necessary by the binary chromosome. Fig. 2 and Fig. 3 illustrate the structure of chromosome for binary and neuroevolutionary approaches, respectively.

2.1 Classifier

It is important to point out that any classifier can be used with the proposed methodology. However, in this study a Support Vector Machine classifier was considered for the experiments.

Support Vector Machines (SVMs) are a set of models with associated learning algorithms that can be applied to classification and regression. The samples in a dataset are represented as points in space, so points of different categories can be separated by a hyper-plane or a set of hyper-planes. Although SVMs are binary linear classifiers, additional methods, such as kernel methods, can be applied to perform non-linear classifications. SVMs classifiers had been successfully applied in many machine learning problems.

The SVM classifier performance heavily depends on the selection of the right parameters, such as kernel function, kernel coefficients and regularization. In this study, a SVM non-linear classifier with Radial Basis Function (RBF) was considered with two different parameters to be optimized: the regularization (C) and the

kernel gamma parameter (γ). This type of classifier was already used by [7] in feature selection problems with multiobjective optimization.

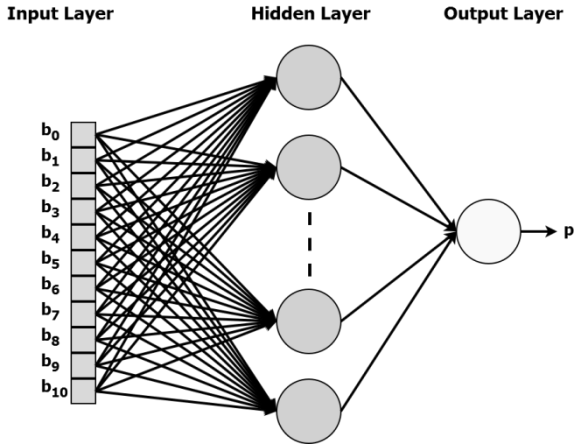

Fig. 1 Neural Network partially represented. Input layer receives a feature number in binary form (bits $b_0, b_1 \ldots b_{10}$). Hidden layer has a total of 20 neurons (only four are show on the figure). Output layer is composed by one single neuron that gives output p, which is the probability of input feature be relevant (selected) to the classifier

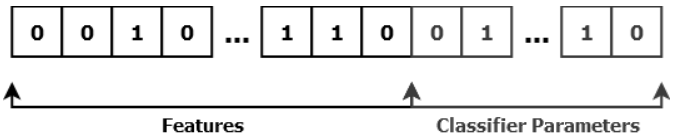

Fig. 2 Example of a chromosome for binary representation. The use information of each feature is encoded in one single bit, parameters for the classifier should be encoded at the end of the chromosome using binary representation

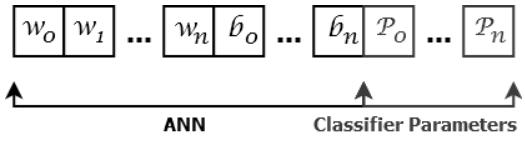

Fig. 3 Chromosome representation for neuroevolutionary approach. Each gene encodes a real number which might represent a weight or bias (of the ANN) or a parameter for the classifier

2.2 Performance Measure for Classification

A systematic analysis of performance measurements for classification can be found in [11]. When dealing with binary classification, *i.e.*, when datasets are composed by samples of two distinct (non-overlapping) classes, the precision metric of the classifier can be expressed by equation:

$$P = \frac{TP}{TP + FP}$$

where TP is the number of true positives, *i.e.*, the number of samples correctly classified and FP is the number of false positives, *i.e.*, the number of samples that belongs to a given class, but were incorrectly assigned to the other class.

For multi-class datasets the precision P can be expressed by the equation:

$$P = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$$

where tp_i is the number of true positives for a given class, fp_i is the number of false positives, *i.e.*, the number of samples of the given class that were incorrectly classified in another class, and l is the total number of possible classes.

2.3 Multiobjective Optimization

In feature selection problems there are two main conflicting objectives: the minimization of the number of features used for classification and the maximization of classifier precision. Thus, multiple solutions with different tradeoffs (number of features versus precision) can emerge from multiobjective optimization approaches.

The methodology proposed in this study combines the reduction of the search space (by using ANNs) with the minimization of objectives (number of features and classification error) into a single approach by using Neuroevolutionary MOEA (Multiobjective Optimization Evolutionary Algorithm). Fig. 4 illustrates the overall algorithm.

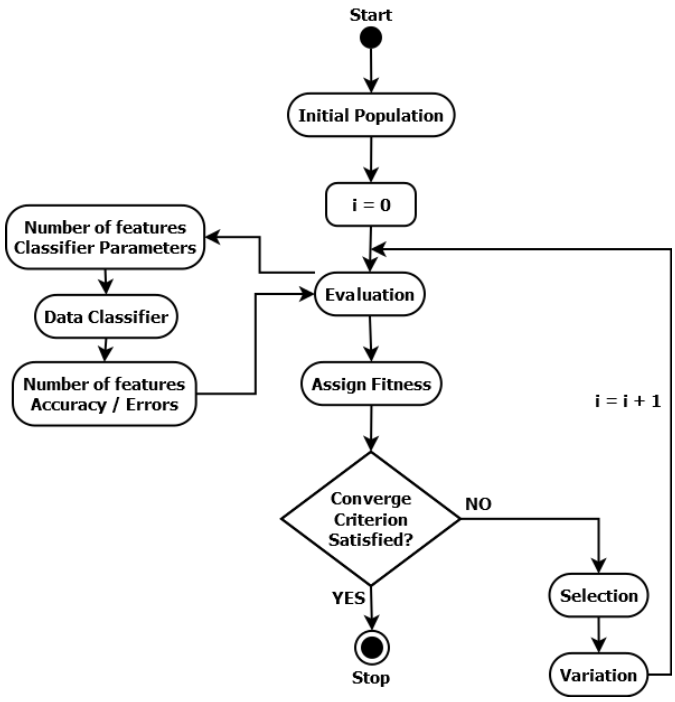

Fig. 4 Algorithm for the proposed approach for feature selection using neuroevolutionary and multiobjective optimization evolutionary methods

The algorithm comprises a multiobjective optimization evolutionary process. It starts by an initial population of solutions which can be randomly generated. The ANNs are used in the evaluation phase to provide the features and parameters to be used by the classifier. The classifier is applied to the dataset considering the provided parameters and objective functions values are calculated from classification results. The process continues by sorting the solutions following a fitness criterion and deciding if convergence is reached or more iterations are needed. Evolution is promoted by selection and variation procedures.

At the end, a Pareto front composed by a set of non-dominated solutions which give different tradeoffs between the number of features used for classification and the precision of the classifier is expected. In this context, two objective functions can be defined:

f_1 = Number of features used for classification

f_2= Classifier error defined as $f_2 = 1 - P$, where P is the classifier precision expressed between [0.0, 1.0].

By defining f_2 as the classifier error, the optimization problem becomes minimize (at the same time) f_1 and f_2.

3 Experimental Design

To evaluate the proposed approach, eight datasets were chosen from UCI Machine Learning Repository[1] and one well known dataset (*colon*) was chosen from the literature in feature selection. All datasets comprise different number of features, samples and classes. Thus, a multiclass SVM classifier implementation was used in the experiments. Table 1 lists all datasets.

Table 1 Datasets used in the experiments

Dataset	Features	Samples	Classes
colon	2000	64	2
ionosphere	34	351	2
musk-1	166	476	2
sonar	60	208	2
semeion	256	1593	10
yeast	8	1484	10
libras	90	360	15
wine1	12	1600	10
solar	12	1066	7

The proposed approach was implemented in MATLAB using the models and functions provided by the Statistics and Machine Learning Toolbox to perform SVM multiclass classification. The multiobjective optimization algorithm was implemented based on the SMS-EMOA algorithm [12]. In each generation, one single offspring is produced. The selection is done using a uniform distribution and variation is performed by the SBX-Crossover operator, which is designed to work with real number representations. Since the parameters of the classifier and of the neural networks are real numbers, this operator is adequate for the neuroevolutionary approach. The fitness of each solution and replacement strategy are based on Pareto front and *hypervolume* measure [13].

To compare the results, a binary approach was also applied to the datasets. The overall algorithm is the same, except by the evaluation and variation phases, where each solution is represented by a binary chromosome (Fig. 2) and a two point crossover operator is used instead of the SBX-Crossover.

[1] Available at https://archive.ics.uci.edu

Concerning the classifier parameters C (regularization) and γ (kernel gamma), after preliminary experiments with all datasets and based on former studies found in the literature, the following intervals were defined: [1, 500] for C and [0.01, 10] for kernel gamma, respectively. To encode these values in the binary representation, 10 bits were used for each parameter. This leads to 2^{10} possible integer values that are normalized into the respective parameter interval.

All classifications were performed using *k*-fold cross- validation with $k = 10$. The partitions for each dataset were pre-defined and used for both binary and neuroevolutionary approaches. The size of each population was set to 150 individuals (solutions) and the number of maximum generations was set to 300 due to computational time constraints.

4 Results and Discussion

Fig. 5 and Fig. 6 show the evolution of the *hypervolume* for each generation for binary and neuroevolutionary approaches, respectively. All values were normalized concerning the origin and the maximum allowed point for all datasets. All curves are visually similar in both cases, but it can be seen that most of the curves in Fig. 6 (neuroevolutionary) converges slightly faster than Fig. 5.

Table 2 lists the *hypervolume* of Pareto front of final populations for both representations. Better results are highlighted. The neuroevolutionary approach presented better results for 5 of the 9 datasets, 3 datasets presented equal results and only one dataset (*wine1*) presented higher *hypervolume* for binary approach.

To illustrate the results of each optimization, Fig. 7 and Fig. 8 show the initial and final populations for datasets *semion* and *colon* (neuroevolutionary), respectively. Other datasets were omitted due to space constraints. It can be seen clearly the evolution of initial population to a set of optimal solutions which gives different tradeoffs between the number of features (f_1) and the classifier error (f_2).

For all datasets, an optimal solution (located in the knee of the Pareto curve) was selected from final population. Table 3 lists these solutions along with its classifier parameters, precision and number of features (better precision results are highlighted). In terms of classifier precision, for five of nine datasets, the neuroevolutionary approach presented better results. For the dataset *sonar*, neuroevolutionary reached 100% of precision using only one feature to classification against the binary approach, which found 2 features with 83% of precision. For datasets *semeion* and *wine1*, the neuroevolutionary approach presented better classifier precision, but the number of features was higher than the binary approach. The results for dataset *semeion* were 85% of precision (neuro) against 83% (binary) and the number of features were 22 (neuro) against 17 (binary). For dataset *wine1*, the results were 75% of precision (neuro) versus 73% (binary) and 4 features (neuro) versus 3 features (binary).

Concerning the dataset *libras*, the neuroevolutionary approach reached 85% of precision against 87% for binary approach, but only 6 features were used (against 7

features for binary). Datasets *ionosphere* and *solar* presented exactly the same results (precision and number of features) for both approaches. Only the dataset *yeast* presented better results for the binary approach: 59% of precision against 58% for neuroevolutionary, using 5 features in both approaches.

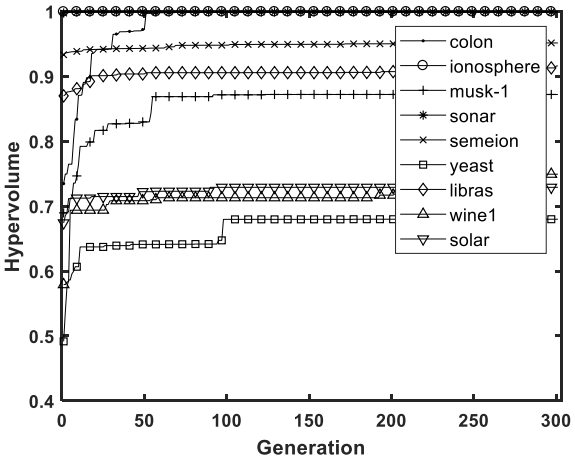

Fig. 5 *Hypervolume* evolution for each dataset using binary representation

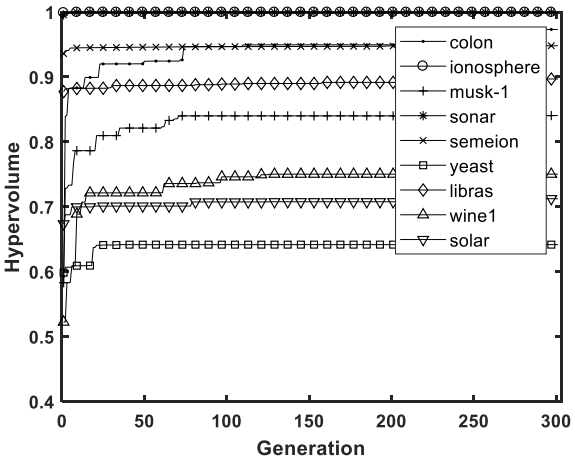

Fig. 6 *Hypervolume* evolution for each dataset using neuroevolutionary approach

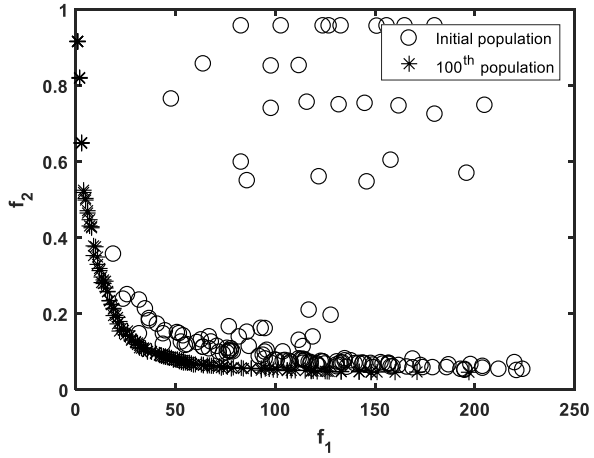

Fig. 7 Initial and final populations for dataset *semeion* (neuroevolutionary approach)

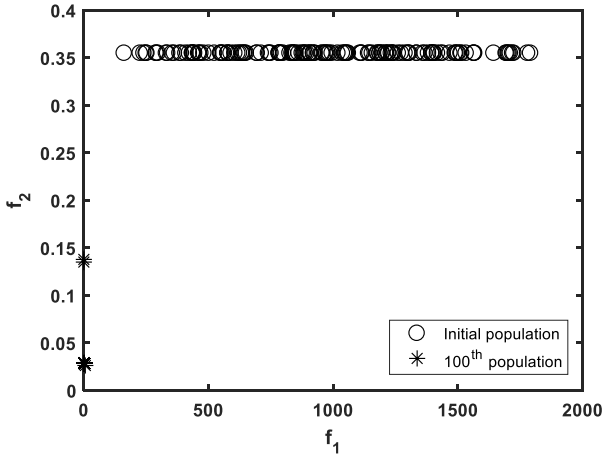

Fig. 8 Initial and final populations for dataset *colon* (neuroevolutionary approach)

Table 2 *Hypervolume* for Pareto front of final populations for binary and neuroevolutionary approaches

	Hypervolume	
Dataset	**Binary**	**Neuroevolutionary**
colon	0.85	0.86
ionosphere	0.99	0.99
musk-1	0.77	0.78
sonar	0.78	0.99
semeion	0.05	0.08
yeast	0.19	0.19
libras	0.22	0.26
wine1	0.50	0.46
solar	0.46	0.46

Table 3 Optimal solutions selected from Pareto front of final population for each dataset (classifier parameters, precision and number of features are listed)

	Binary			Neuroevolutionary		
Dataset	**C, γ**	**P**	**f_s**	**C, γ**	**P**	**f_s**
colon	324.08, 8.33	0.97	2	45.07, 8.24	0.98	2
ionosphere	17.08, 0.47	1.00	1	354.72, 9.99	1.00	1
musk-1	32.19, 9.71	0.82	2	124.01, 18.57	0.84	2
sonar	90.67, 0.63	0.83	2	72.99, 3.00	1.00	1
semeion	258.30, 1.58	0.83	17	474.86, 0.89	0.85	22
yeast	1.00, 0.16	0.59	5	475.85, 1.89	0.58	5
libras	1.00, 0.33	0.87	7	288.02, 0.20	0.85	6
wine1	23.90, 0.01	0.72	3	218.43, 0.02	0.75	4
solar	21.47, 2.52	0.71	3	126.47 ,3.31	0.71	3

Table 4 shows the features that correspond to the optimal solutions obtained using the neuroevolutionary and binary approaches for the *colon* dataset. The precision, number of features and features selected in each solution are indicated. It can be observed that the number of solutions and the number of features of each solution using the neuroevolutionary approach are smaller. Feature 1 is present in all solutions. Feature 513 is selected for 2 neuroevolutionary solutions and 5 binary solutions. Features 2001, 2003, 2005, 2008, 2010, 2011, 2015, 2019 and 2020 are present in binary solutions. Solutions B6 to B10 have a precision of 1.000 and are very similar, sharing a large number of features.

Table 4 Optimal solutions from the final population for dataset *colon*

	Neuroevolutionary			Binary									
P	0.865	0.971	0.974	0.854	0.971	0.972	0.973	0.974	1.000	1.000	1.000	1.000	1.000
f_i	2	3	7	14	15	16	21	22	23	24	24	22	22
Feature	**N1**	**N2**	**N3**	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**
780	■												
1184		■											
769			■										
773			■										
785			■										
833			■										
837			■										
853			■										
513		■	■	■	■	■	■	■					
1	■	■	■	■	■	■	■	■	■	■	■	■	■
2001				■	■	■	■	■	■	■	■	■	■
2003				■	■	■	■	■	■	■	■	■	■
2005				■	■	■	■	■	■	■	■	■	■
2008				■	■	■	■	■	■	■	■	■	■
2010				■	■	■	■	■	■	■	■	■	■
2011				■	■	■	■	■	■	■	■	■	■
2015				■	■	■	■	■	■	■	■	■	■
2019				■	■	■	■	■	■	■	■	■	■
2020				■	■	■	■	■	■	■	■	■	■
2002				■	■	■	■	■	■	■	■	■	
2013				■	■	■	■	■	■	■	■		■
1740						■	■	■	■	■	■	■	■
42							■	■	■	■	■	■	■
187							■	■	■	■	■	■	■
498							■	■	■	■	■	■	■
1955							■	■	■	■	■	■	■
2007							■	■	■	■	■	■	■
544									■	■	■	■	■
632									■	■	■	■	■
1464									■	■	■	■	■
1466									■	■	■	■	■
1497									■	■	■	■	■
2017				■	■	■					■		
883							■	■					
1483					■	■							
1687							■	■					
102								■					
2004										■			

5 Conclusions

This study proposes a neuroevolutionary approach to deal with feature selection problems by using multiobjective evolutionary algorithms. Considering *n*-dimensional datasets, to perform feature selection using binary representations or exhaustive search becomes impractical for a large *n*. In this context, the proposed approach can drastically reduce the search space by using Artificial Neural Networks to provide the most important features to classify the data with maximum precision. Since the number of features and the classification precision are conflicting objectives, by using multiobjective optimization a set of solutions (Pareto front) with different tradeoffs between the objectives can be obtained.

The methodology was applied to nine datasets with different number of features, samples and classes. To compare the results, a binary representation was also applied. When comparing the Pareto front of both representations (in terms of *hypervolume*), the neuroevolutionary approach presented better (or equal) results for eight of nine datasets.

For each dataset, an optimal solution was selected from the Pareto front considering the point closest to the knee of the curve (to give an equal relationship between classifier precision and the number of features). When comparing these points in both representations, for seven of nine datasets the neuroevolutionary approach presented better (or equal) results in terms of classifier precision. Different results were also achieved for the number of features. Only one dataset presented better results for binary approach. However, it is important to point out that by using the neuroevolutionary approach, the search space is drastically reduced, since the parameters of ANN are being evolved instead of the binary representation for each feature.

By including classifier parameters in the optimization, the algorithm was able to find the best combination of C (regularization) and kernel gamma (of the SVM Classifier) for each dataset in order to reach better classification precision.

Future works can address different parameters or kernel functions for the SVM classifier, or even the use of other classifiers to perform the classification. Other ANN topologies can also be considered.

Acknowledgments This work has been supported by FCT - Fundação para a Ciência e Tecnologia in the scope of the projects: PEst-OE/EEI/UI0319/2014, UID/MAT/00013/2013, UID/CEC/00319/2019 and the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

1 I. Guyon, S. Gunn, M. Nikravesh and L. A. Zadeh, Feature extraction: foundations and applications, vol. 207, Springer, 2008.

2 A. Unler, A. Murat and R. B. Chinnam, "mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification," *Information Sciences,* vol. 181, pp. 4625-4641, 2011.

3 E. Hancer, B. Xue, M. Zhang, D. Karaboga and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Information Sciences,* vol. 422, pp. 462-479, 2018.

4 J. Bi, "Multi-objective programming in SVMs," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003.

5 C. Igel, "Multi-objective model selection for support vector machines," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2005.

6 L. S. Oliveira, M. Morita and R. Sabourin, "Feature selection for ensembles using the multi-objective optimization approach," in *Multi-Objective Machine Learning*, Springer, 2006, pp. 49-74.

7 A. Gaspar-Cunha, "Feature selection using multi-objective evolutionary algorithms: application to cardiac SPECT diagnosis," in *Advances in Bioinformatics*, Springer, 2010, pp. 85-92.

8 R. Pinto, H. Silva, F. Duarte, J. Nunes and A. Gaspar-Cunha, "Neuroevolutionary Multiobjective Methodology for the Optimization of the Injection Blow Molding Process," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2019.

9 R. Denysiuk, F. M. Duarte, J. P. Nunes and A. Gaspar-Cunha, "Evolving neural networks to optimize material usage in blow molded containers," *EUROGEN - International Conference on Evolutionary and Deterministic Methods for Design Optimization and Control with Applications to Industrial and Societal Problems,* 2017.

10 R. Denysiuk, A. Gaspar-Cunha and A. C. B. Delbem, "Neuroevolution for solving multiobjective knapsack problems," *Expert Systems with Applications,* vol. 116, pp. 65-77, 2019.

11 M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management,* vol. 45, pp. 427-437, 2009.

12 N. Beume, B. Naujoks and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research,* vol. 181, pp. 1653-1669, 2007.

13 E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *international conference on parallel problem solving from nature*, 1998.